Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (Currently Amended) A computer-implemented software application framework embodied on a computer comprising:

a virtual object space providing access to a plurality of objects, each object having a set of functionality and being identifiable by a unique identifier provided by the virtual object space, and providing generic object functionality for the plurality of objects including an associations and transactions functionality for relating the plurality of objects and interaction between the plurality of objects, the associations and transactions functionality configured to provide transactions using a two-phase commit to handle different memory copies of object states of the plurality of objects, a distribution functionality for locking, flushing, and copying of the virtual object space, and a persistency functionality for maintaining persistency of the plurality of objects, the generic object functionally including a first interface for applications using the plurality of objects and a second interface for a service deploying the plurality of objects, the service providing an adapter to map the unique identifier to the plurality of objects and to objects internal to the service, the access being a virtual access provided using the unique identifier without physically storing the plurality of objects at the virtual object space; and a visualization framework for visualizing the plurality of objects independently of an application implementing the object, each object being visualized using the object's unique identifier and according to the object's corresponding set of functionality.

2. (Canceled).

3. (Canceled).

4. (Previously Presented) The computer-implemented framework of claim 1, further comprising:

framework services providing common event handling of the plurality of objects.

5. (Currently Amended) A computer-implemented object browser computer framework embodied on a computer comprising:

a virtual object space providing access to a plurality of objects, each object being identifiable by a unique identifier, and providing generic object functionality for the plurality of objects including an associations and transactions functionality for relating the plurality of objects and interaction between the plurality of objects, the associations and transactions functionality configured to provide transactions using a two-phase commit to handle different memory copies of object states of the plurality of objects, a distribution functionality for locking, flushing, and copying of the virtual object space, and a persistency functionality for maintaining persistency of the plurality of objects, the generic object functionally including a first interface for applications using the plurality of objects and a second interface for a service deploying the plurality of objects, the service providing an adapter to map the unique identifier to the plurality of objects and to objects internal to the service, the access being a virtual access provided using the unique identifier without physically storing the plurality of objects at the virtual object space;

a user interface for receiving input from a user relating to a desired arrangement of one or more of the plurality of objects; and

a visualization framework for visualizing objects according to the desired arrangement, each object being visualized using the object's unique identifier independently of an application implementing the object.

6. (Previously Presented) The computer-implemented object browser framework of claim 5, wherein the user interface further includes allowing a user to browse the plurality of objects independent of one or more applications implementing the objects.

7. (Previously Presented) The computer-implemented object browser framework of claim 5, wherein the desired arrangement is hierarchical.

8. (Previously Presented) The computer-implemented object browser framework of claim 5, wherein the visualization framework includes an object viewer.

9. (Currently Amended) A method of facilitating software object browsing, the method comprising:

providing access to software objects through a virtual object space in which each software object has an associated set of functionality and is identifiable by a unique identifier, and providing generic object functionality for software objects including an associations and transactions functionality for relating the software objects and interaction between the software objects, the associations and transactions functionality configured to provide transactions using a two-phase commit to handle different memory copies of object states of the plurality of objects, a distribution functionality for locking, flushing, and copying of the virtual object space, and a persistency functionality for maintaining persistency of the software objects, the generic object functionally including a first interface for applications using the plurality of objects and a second interface for a service deploying the plurality of objects, the service providing an adapter to map the unique identifier to the plurality of objects and to objects internal to the service, the access being a virtual access provided using the unique identifier without physically storing the plurality of objects at the virtual object space; and

visualizing objects independently of an application implementing the software objects, each object being visualized using the object's unique identifier and according to the object's corresponding set of functionality.

10. (Canceled).

11. (Canceled).

12. (Original) The method of claim 9, further comprising:

providing framework services providing common event handling of the software objects.

13. (Previously Presented) The computer-implemented framework of claim 1, further comprising:

providing a portlet to access the generic object functionality to visualize the structure of the plurality of objects.

14. (Previously Presented) The method of claim 9, further comprising:

providing a portlet to access the generic object functionality to visualize the structure of the software objects.

15. (Canceled).

16. (Previously Presented) The method of claim 9, further comprising:

implementing the plurality of objects to include the first interface accessible by the framework and the second interface accessible by a service acting as an adapter between the framework and an application separate from the framework.